Available online at www.sciencedirect.com

ScienceDirect

Finite Fields and Their Applications 14 (2008) 1068–1082

FINITE FIELDS AND THEIR APPLICATIONS

http://www.elsevier.com/locate/ffa

On the qth power algorithm

Xiaochun Hu, Hiren Maharaj [*,1]

Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975, USA

Received 5 February 2008

Available online 23 July 2008

Communicated by Gary L. Mullen

Abstract

Leonard and Pellikaan developed the qth power algorithm to compute module bases for the integral closure of the polynomial ring $\mathbb{F}_q[x]$ in a class of function fields. In this paper, their algorithm is adapted to efficiently obtain an $\mathbb{F}_q$-basis for a class of Riemann–Roch spaces without having to compute the entire integral closure. This reformulation allows one to determine the complexity of the algorithm. Further, we obtain a simple characterization of the integral closure.

Keywords: Riemann–Roch spaces; Integral closure; Explicit bases; Algebraic geometric codes

1. Introduction

Explicit bases for Riemann–Roch spaces of divisors from function fields over finite fields are necessary in order to explicitly construct algebraic geometric codes. Leonard [6] invented the qth power algorithm to compute integral closures of $\mathbb{F}_q[x]$ in two towers of function fields introduced by Garcia and Stichtenoth [2,3]. Leonard and Pellikaan [7] adapted this algorithm, using Groebner basis techniques, to compute a module basis for the integral closure of $\mathbb{F}_q[x]$ in a generalized version of curves and surfaces of type I. In particular, they compute a module basis for the integral closure of $\mathbb{F}_q[x]$ in the function field $F = \mathbb{F}_q(x, y)$ defined by $f(y) = 0$ where

* Corresponding author.
E-mail addresses: xiaochh@clemson.edu (X. Hu), hmahara@clemson.edu (H. Maharaj).

[1] Supported in part by NSA grant MDA904-05-1-0046.

1071-5797/$ – see front matter
doi:10.1016/j.ffa.2008.06.005

$f(T)$ is a polynomial of the form

$$f(T) = x^a + T^b + g(x, T) \quad \text{where } \gcd(a, b) = 1 \text{ and } a > b > \deg g. \tag{1}$$

Under these assumptions, $f(T)$ is irreducible [4, Example 3.16], the pole P_∞ of x in $\mathbb{F}_q(x)$ is totally ramified in F. If Q_∞ is the unique place of F which lies above P_∞, then the integral closure of $\mathbb{F}_q[x]$ in F can be written as the union $\bigcup_{m=1}^{\infty} \mathcal{L}(mQ_\infty)$ of Riemann–Roch spaces. Thus one obtains $\mathbb{F}_q$-bases for the Riemann–Roch spaces $\mathcal{L}(mQ_\infty)$.

Next we describe the main results in this paper. Let $F := \mathbb{F}_q(x, y)$ be a field extension of $\mathbb{F}_q(x)$ defined by $f(y) = 0$ where $f(T) \in (\mathbb{F}_q[x])[T]$ is monic, separable and irreducible. Throughout we let $R := \mathbb{F}_q[x]$ and we denote the integral closure of R in F by $\mathrm{ic}(R)$. Let $D(x) \in \mathbb{F}_q[x]$ be the unique monic polynomial of smallest degree such that $\frac{1}{f'(y)} = \frac{g(y)}{D(x)}$ where $g(y) \in \mathbb{F}_q[x, y]$ and $f'(y)$ is the formal derivative of f evaluated at y. In Section 2, we obtain the following characterization of the integral closure of $\mathbb{F}_q[x]$ in F which is an improvement of a similar result by Mattig [8]. Let $\mathfrak{Q}$ be any integer greater than $\bar{\mathfrak{Q}} := b\ell$ where ℓ is the largest exponent in the factorization of $D(x) \in \mathbb{F}_q[x]$ as a product of irreducible polynomials.

Theorem 3. *For $f \in \mathbb{F}_q[x, y]/D \subset F$,*

$$f \in \mathrm{ic}(R) \quad \textit{if and only if} \quad f^{\mathfrak{Q}} \in \mathbb{F}_q[x, y]/D.$$

In [8, p. 15], the same result is proved but with a larger number for $\bar{\mathfrak{Q}}$. An important consequence of Theorem 3 for our purposes is that the qth power algorithm terminates in at most t rounds where t is the smallest integer such that $q^t > b\ell$.

Next we specialize to the function field F defined by Eq. (1). In Section 3 we reformulate the qth power algorithm so that one efficiently obtains an $\mathbb{F}_q$-basis for the spaces $\mathcal{L}(mQ_\infty)$ without having to compute the entire integral closure. In Section 4 we present a simple example. Furthermore, we compute the complexity (in Section 5) of the qth power algorithm when applied to functions fields of type I. We prove

Theorem 14. *The complexity of the qth power algorithm to compute a basis of $\mathcal{L}(mQ_\infty)$ is*

$$\mathrm{O}\big(q^3 a^3 b^9 m^3 \log_q(ab^3)\big). \tag{2}$$

Note that throughout this paper we use the notation of [11].

2. Preliminary results

In this section we prove some general results regarding integral closures. Let $F := \mathbb{F}_q(x, y)$ be a field extension of $\mathbb{F}_q(x)$ such that $f(y) = a_b + a_{b-1}y + \cdots + y^b = 0$, where $a_i \in F_q[x]$ and $f(T) = a_b + a_{b-1}T + \cdots + T^b$ is separable and irreducible. We put $R = \mathbb{F}_q[x]$. We denote the R module generated by $f_1, f_2, \ldots, f_j \in F$ by $\langle f_1, f_2, \ldots, f_j \rangle_R$. Clearly, there exist a unique monic polynomial $D(x) \in \mathbb{F}_q[x]$ of smallest degree and a polynomial $g(y) \in (\mathbb{F}_q[x])[y]$ such that $\frac{1}{f'(y)} = \frac{g(y)}{D(x)}$. For example, if $f(y) = y^3 + xy + x^5$ over $\mathbb{F}_2$ then $f'(y) = y^2 + x$ so $1/f' = 1/(y^2 + x) = y/(y^3 + yx) = y/x^5$ so $D(x) = x^5$.

Theorem 1. *(See [5].) Put* $f_j(y) = \sum_{i=0}^{j} a_i y^{j-i}$, $0 \leqslant j \leqslant b-1$, *where* $a_0 = 1$. *Then the integral closure of* $\mathbb{F}_q[x]$ *in* $\mathbb{F}_q(x, y)$ *is contained in the* $\mathbb{F}_q[x]$ *module generated by*

$$\frac{f_{b-1}(y)}{f'(y)}, \frac{f_{b-2}(y)}{f'(y)}, \ldots, \frac{f_0(y)}{f'(y)}.$$

In particular, the integral closure of $\mathbb{F}_q[x]$ *is contained in*

$$\left\langle \frac{1}{D(x)}, \frac{y}{D(x)}, \ldots, \frac{y^{b-1}}{D(x)} \right\rangle = \frac{1}{D(x)} \mathbb{F}_q[x, y] \subset F.$$

Put

$$V_0 := \left\langle \frac{1}{D}, \frac{y}{D}, \ldots, \frac{y^{b-1}}{D} \right\rangle_R. \tag{3}$$

For $i \geqslant 1$, put

$$W_i = \left\{ v \in V_0 \;\middle|\; v^i \in V_0 \right\}. \tag{4}$$

Since $\mathrm{ic}(R)$ is a ring, if $\alpha \in \mathrm{ic}(R)$, then $\alpha^i \in \mathrm{ic}(R)$. Since $\mathrm{ic}(R) \subseteq V_0$, it follows that $\mathrm{ic}(R) \subseteq W_i$ for all i.

Theorem 2.

(i) *Suppose that* $D(x) = f_1(x)^{\ell_1} f_2(x)^{\ell_2} \cdots f_t(x)^{\ell_s}$ *is a factorization of* D *as a product of irreducible polynomials and let* ℓ *be the maximum of the exponents* $\ell_1, \ell_2, \ldots, \ell_s$. *Then for* $i > b\ell$, $W_i = \mathrm{ic}(R)$.

(ii) *Suppose that none of the zeroes of* D *ramify in* F. *Then for* $i > \ell$ *we have that* $W_i = \mathrm{ic}(R)$. *In particular, if* $\ell = 1$ *then* $W_2 = \mathrm{ic}(R)$.

Proof. (i) Suppose that $Q_1, Q_2, \ldots, Q_j$ is the set of places of F which lie above P_∞. Then

$$\mathrm{ic}(R) = \bigcup_{m_1, m_2, \ldots, m_j = 1}^{\infty} \mathcal{L}(m_1 Q_1 + \cdots + m_j Q_j)$$

(see [11, Theorem III.2.6]). The functions in V_0 belong to

$$\Lambda := \bigcup_{m_1, m_2, \ldots, m_j = 1}^{\infty} \mathcal{L}\big(m_1 Q_1 + \cdots + m_j Q_j + (D)_0\big)$$

where $(D)_0$ is the zero divisor of D in F. The functions f in W_2 have the property that $f^2 \in V_0 \subseteq \Lambda$. Let Q be a place in the support of $(D)_0$ (which is a divisor in F). Then the place $P = Q \cap \mathbb{F}_q(x)$ is the unique zero of one of the polynomials $f_i(x)$. Now $v_Q(f^2) \geqslant -v_Q((D)_0)$ whence

$$v_Q(f) \geqslant \frac{-1}{2} v_Q\big((D)_0\big) = \frac{-1}{2} e(Q|P)\ell_i \geqslant \frac{-1}{2} b\ell$$

where $e(Q|P)$ is the ramification index of Q. Thus

$$W_2 \subseteq \bigcup_{m_1,m_2,\ldots,m_j=1}^{\infty} \mathcal{L}\left(m_1 Q_1 + \cdots + m_j Q_j + \frac{1}{2}(D)_0\right)$$

(we are allowing the coefficients of places in a divisor to be rational numbers; it should be clear what is meant). Likewise, in general we have that

$$W_i \subseteq \bigcup_{m_1,m_2,\ldots,m_j=1}^{\infty} \mathcal{L}\left(m_1 Q_1 + \cdots + m_j Q_j + \frac{1}{i}(D)_0\right)$$

for $i = 1, 2, \ldots$ and so for $f \in W_i$ we have

$$v_Q(f) \geqslant \frac{-1}{i} v_Q\big((D)_0\big) = \frac{-1}{i} e(Q|P)\ell_i \geqslant \frac{-b\ell}{i}.$$

It follows that if $i > b\ell$ then

$$W_i \subseteq \bigcup_{m_1,m_2,\ldots,m_j=1}^{\infty} \mathcal{L}(m_1 Q_1 + \cdots + m_j Q_j) = \mathrm{ic}(R).$$

The proof of (ii) now follows likewise. □

We obtain the following characterization of the integral closure of $\mathbb{F}_q[x]$. Let $\bar{\Omega} := b\ell$. Let Ω be any integer greater than $\bar{\Omega}$.

Theorem 3. *For $f \in V_0$,*

$$f \in \mathrm{ic}(R) \quad \textit{if and only if} \quad f^{\Omega} \in V_0$$

with ℓ as defined in Corollary 5.

A similar result is proved in [8]. Below we make a comparison, but first we state this result from [8]. Suppose $w_1, \ldots, w_b$ is a basis of F as a vector space over $\mathbb{F}_q(x)$ such that $\mathrm{ic}(R) \subseteq \mathbb{F}_q[x]w_1 + \mathbb{F}_q[x]w_2 + \cdots + \mathbb{F}_q[x]w_b =: M$. Let Ω be any integer greater than $\bar{\Omega} := \max\{-v_{Q_i}(w_k)\colon\ 1 \leqslant i \leqslant j \text{ and } 1 \leqslant k \leqslant b\}$. Then

Theorem 4. *(See Mattig [8].) For $f \in M$,*

$$f \in \mathrm{ic}(R) \quad \textit{if and only if} \quad f^{\Omega} \in M.$$

Let $d_x = \max(\deg(a_i)\colon\ 1 \leqslant i \leqslant b)$. In our situation, we take $w_k = y^k/D(x)$. Now observe that $-v_{Q_i}(y^k/D(x)) = k(-v_{Q_i}(y)) + de(Q_i|P_\infty) \leqslant bd_x + bd$. Thus Theorem 4 implies that $\bar{\Omega} = b(d + d_x)$ while Theorem 3 gives a smaller number for this quantity, namely, $\bar{\Omega} = b\ell$.

We are interested in applications to the qth power algorithm. For this purpose, for $i \geqslant 1$, one defines

$$V_i = \left\{ v \in V_{i-1} \;\middle|\; v^q \in V_{i-1} \right\}. \tag{5}$$

We have the following corollary of Theorem 2.

Corollary 5.

(i) *If t is the smallest integer such that $q^t > b\ell$, then for $i \geqslant t$ we have that $V_i = \mathrm{ic}(R)$.*
(ii) *Suppose that none of the zeroes of D ramify in F. If t is the smallest integer such that $q^t > \ell$, then for $i \geqslant t$ we have that $V_i = \mathrm{ic}(R)$. In particular, if $\ell = 1$ then $V_2 = \mathrm{ic}(R)$.*

In the qth power algorithm [7], module bases are computed for each successive V_i. This result implies that the algorithm terminates once V_t is computed, that is, after t rounds.

3. qth power algorithm to find an $\mathbb{F}_q$-basis

The following result is used in [7]. We present the proof since we cannot find a reference.

Proposition 6. *Assume $F := \mathbb{F}_q(x, y)$ is the function field defined by*

$$f(y) = x^a + y^b + g(x, y) = 0 \tag{6}$$

where $\gcd(a, b) = 1$ and $a > b > \deg g$. Let P_∞ denote the pole of x in $\mathbb{F}_q(x)$. Then P_∞ is totally ramified in the extension $\mathbb{F}_q(x) \subset F$. If Q_∞ denote the unique place of F which lies above P_∞, then the pole order of Q_∞ at x and y are b and a, respectively.

Proof. The fact that $f(T)$ is irreducible follows from the conditions [4, Example 3.16] that $\gcd(a, b) = 1$ and $a > b > \deg g$. Let Q be any place of F which lies above P_∞ and let v denote the discrete valuation corresponding to Q. We claim that $av(x) = bv(y)$. The main result follows from this: let $e := e_{Q|P_\infty} \leqslant b$. Then $v(x) = e \cdot v_{P_\infty}(x) = -e$. By the claim, b divides $av(x)$. It follows that b divides $v(x) = -e$, as $\gcd(a, b) = 1$. Then $e \leqslant b$ implies that $e = b$ and so P_∞ is totally ramified in F, $v(x) = -b$ and $v(y) = -a$. Next we prove the claim. First, note that $v(x) = -e < 0$. Suppose $av(x) > bv(y)$. As $v(x^a + y^b) = v(g(x, y))$, we have $bv(y) = v(\sum_{a_{i,j} \neq 0} a_{i,j} x^i y^j) \geqslant v(x^i y^j) = iv(x) + jv(y)$ for some i and j, where $i + j < b$. Since $v(x)$ is negative and $av(x) > bv(y)$, $v(y)$ is negative as well. Hence, $b - j \leqslant \frac{bi}{a}$. It follows that $(b - j)v(y) \geqslant iv(x) > \frac{bi}{a} v(y)$ (note that it is not possible to have $i = 0$). As $i < b - j$, we have $ai < a(b - j) < bi$, which implies $a < b$, a contradiction. Now suppose $av(x) < bv(y)$. As $v(x^a + y^b) = v(g(x, y))$, we have $av(x) = v(\sum_{a_{i,j} \neq 0} a_{i,j} x^i y^j) \geqslant v(x^i y^j) = iv(x) + jv(y)$ for some i and j, where $i + j < b$. It follows that $(a - i)v(x) \geqslant jv(y) > \frac{aj}{b} v(x)$. Since $v(x)$ is negative, $a - i < \frac{aj}{b}$, that is, $a(b - j) < bi$, which implies $a < b$, a contradiction. Therefore, $av(x) = bv(y)$ as required. □

Henceforth we will work with the function field F as defined in Proposition 6. Also the notation of Proposition 6 will be used throughout this paper. We define V_i $(i \geqslant 0)$ as in (3) and (5).

Lemma 7. *Suppose that $x^{i_1}y^{j_1}$ and $x^{i_2}y^{j_2}$ have the same pole order at Q_∞. If $0 \leqslant j_1, j_2 \leqslant b-1$ then $i_1 = i_2$ and $j_1 = j_2$.*

Proof. Let v denote the discrete valuation corresponding to Q_∞. We claim that $av(x) = bv(y)$. By assumption, $v(x^{i_1}y^{j_1}) = v(x^{i_2}y^{j_2})$. It follows that $i_1 b - j_1 v(y) = i_2 b - j_2 v(y)$ whence $j_1 v(y) \equiv j_2 v(y) \bmod b$ so that b divides $(j_1 - j_2)v(y)$. Since b and $v(y)$ are relatively prime by assumption, it follows that b divides $j_1 - j_2$. Since $0 \leqslant j_1, j_2 \leqslant b-1$ it follows that $j_1 = j_2$ and so we also have $i_1 = i_2$. □

Since $f(T) \in \mathbb{F}_q[x, T]$ is monic in T, any function $h(x, y) \in \mathbb{F}_q[x, y] \subset F$ can be uniquely written in the form $\alpha_0 + \alpha_1 y + \cdots + \alpha_{b-1}y^{b-1}$ where $\alpha_0, \alpha_1, \ldots, \alpha_{b-1} \in \mathbb{F}_q[x]$. Henceforth it will be understood that all functions of $\mathbb{F}_q[x, y]$ $(\subset F)$ are written in this way. By the leading term of $h(x, y)$ we mean the term $cx^i y^j$ $(c \in \mathbb{F}_q \setminus \{0\}, 0 \leqslant j \leqslant b-1)$ of the largest pole order (this term is unique by Lemma 7). We denote this term by $\mathrm{LT}(h)$.

We are ready to describe the algorithm. For clarity, the first round is explained in detail.

Arrange the monomials $x^i y^j$ $(0 \leqslant j \leqslant b-1$ and $i = 0, 1, 2, \ldots)$ in increasing pole order. Denote these by $g_1^{(0)}, g_2^{(0)}, \ldots$.

Put $g_i := g_i^{(0)}$ for $i = 1, 2, 3, \ldots$ (we introduce this notation since the functions g_i will change, yet the $g_i^{(0)}$ will not change). Note that these functions form an $\mathbb{F}_q$-basis for DV_0 and they have distinct pole orders at Q_∞. Let $\mathcal{P}_0$ denote the set of pole orders at Q_∞ of the functions $D^{q-1}g_i^{(0)}$. The numbers in $\mathcal{P}_0$ are of the form $(q-1)d + ia + jb$ where $i \geqslant 0$ and $0 \leqslant j \leqslant b-1$. The largest integer not representable in this form [12] is $(q-1)d + ab - a - b$ so every integer greater than this number belongs to $\mathcal{P}_0$. Then there exist unique $a_i \in \mathbb{F}_q$ and $r \in \mathbb{F}_q[x, y]$ such that

$$g = \sum_i a_i D^{q-1} g_i^{(0)} + r$$

where the pole order of every term of $r \in \mathbb{F}_q[x, y]$ does not belong to $\mathcal{P}_0$. This follows since the functions $D^{q-1}g_i$ also have distinct pole orders. We denote the function r by $NF_0(g)$ and we refer to r as the normal form of g *with respect to* $g_1^{(0)}, g_2^{(0)}, \ldots$. Note that NF_0 is an $\mathbb{F}_q$-linear map.

Put $r_i := NF_0(g_i^q)$ for $i = 1, 2, 3, \ldots$. This set of functions $r_1, r_2, r_3, \ldots$ is *finite* as their pole orders are all bounded by $(q-1)bd + ab - a - b$. Suppose $\mathrm{LT}(r_i) = a_i x^{b_i} y^{c_i}$ (a_i is a nonzero element of $\mathbb{F}_q$) for $i = 1, 2, \ldots$. If r_i and r_j have the same pole order, then by Lemma 7 it follows that $b_i = b_j$ and $c_i = c_j$.

For each $i = 2, 3, \ldots$ and each $1 \leqslant j < i$ we perform the following operation on the r_i for as long as possible.

Gaussian reduction.

(∗) *If the pole order of r_i equals that of r_j for any $1 \leqslant j < i$, then replace r_i by $r_i - \frac{a_i}{a_j} r_j$ and replace g_i by $g_i - \frac{a_i}{a_j} g_j$.*

Note that for a given $i > 1$, it may be necessary to scan through the functions $r_1, r_2, \ldots, r_{i-1}$ several times until the above operation is no longer applicable. Since the functions r_i form a finite set, there exists an N such that $r_i = 0$ for all $i > N$.

The final sets of functions $g_1, g_2, \ldots$ and $r_1, r_2, \ldots$ have the following properties:

(i) They form an $\mathbb{F}_q$-basis for DV_0 because the linear algebra operations are invertible.
(ii) Their qth powers also form an $\mathbb{F}_q$-basis for $D^q V_0^q$ for the same reason as above.
(iii) The nonzero functions among $r_1, r_2, \ldots, r_N$ have distance pole orders at Q_∞ and so are $\mathbb{F}_q$-linearly independent.
(iv) The operation $(*)$ does not change the pole order at Q_∞, so the pole orders of the $g_i^{(0)}$ equals the pole orders of the g_i. This is because, in order to compute g_i, we subtract multiples of the functions $g_j^{(0)}$ (which have lower pole orders) from $g_i^{(0)}$ for $j < i$.

Theorem 8. *Let B be the set of functions g_j with $r_j = NF_0(g_j^q) = 0$. Then B forms an $\mathbb{F}_q$-basis for DV_1.*

Proof. Note that $B \subset DV_1$. The functions in B are linearly independent by (i) above. We must show that they span DV_1. Let $g \in DV_1$. Then $\frac{g}{D} \in V_1$ and so $(\frac{g}{D})^q \in V_0$. Also $\frac{g}{D} \in V_0$.

Since the functions g_i form an $\mathbb{F}_q$-basis for DV_0, there are $a_i \in \mathbb{F}_q$ such that $g = \sum_i a_i g_i^{(0)}$. Now, there exist unique $b_{i,j} \in \mathbb{F}_q$ such that

$$\left(g_i^{(0)}\right)^q = \sum_j b_{i,j} D^{q-1} g_j^{(0)} + r_i$$

where $r_i = NF_0(g_i^q)$. So

$$g^q = \sum_{i,j} a_i b_{i,j} D^{q-1} g_j^{(0)} + \sum_i a_i r_i,$$

whence

$$\left(\frac{g}{D}\right)^q = \sum_{i,j} a_i b_{i,j} \frac{g_j^{(0)}}{D} + \sum_i a_i \frac{r_i}{D^q}.$$

Since $(\frac{g}{D})^q$ and the $\frac{g_j^{(0)}}{D}$ belong to V_0, it follows that $\sum_i a_i \frac{r_i}{D^q} \in V_0$, whence $\sum_i a_i r_i \in \mathrm{Span}\{D^{q-1} g_i^{(0)} \colon i \geqslant 1\}$. But this is impossible since the pole order of $\sum_i a_i r_i$ cannot belong to the set $\mathcal{P}_0$. □

Next we describe round 2 of the algorithm. Discard the functions g_i for which $r_i \neq 0$. We call these functions $g_1^{(1)}, g_2^{(1)}, \ldots$, numbered in the order of increasing pole order. Let $\mathcal{P}_1$ denote the set of pole orders of the functions $D^{q-1} g_1^{(1)}, D^{q-1} g_2^{(1)}, \ldots$ at Q_∞. Note that $\mathcal{P}_1 \subset \mathcal{P}_0$. As in round one, we also put $g_i := g_i^{(1)}$ for $i \geqslant 1$ (the functions g_i change during the round, while the functions $g_i^{(1)}$ are fixed throughout). These functions form an $\mathbb{F}_q$-basis for DV_1.

Then there exist unique $a_i \in \mathbb{F}_q$ and $r \in \mathbb{F}_q[x, y]$ such that

$$g = \sum_i a_i D^{q-1} g_i^{(1)} + r$$

where the pole order of *every term* of $r \in \mathbb{F}_q[x, y]$ does not belong to $\mathcal{P}_1$. This follows since the functions $D^{q-1} g_i^{(1)}$ all have distinct pole orders. We denote the function r by $NF_1(g)$ and we refer to r as the normal form of g *with respect to* $g_1^{(1)}, g_2^{(1)}, \ldots$. Note that NF_1 is an $\mathbb{F}_q$-linear map. Put

$$r_i := NF_1\left(g_i^q\right) \quad \text{for } i = 1, 2, 3, \ldots.$$

This sequence of functions $r_1, r_2, r_3, \ldots$ is *finite* as their pole orders are all bounded because $\mathbb{N}_0 \setminus \mathcal{P}_1$ is finite. We perform the operations $(*)$ on the r_i's as before. Since the functions r_i form a finite set, there exists an N such that $r_i = 0$ for all $i > N$. The final sets of functions $g_1, g_2, \ldots$ and $r_1, r_2, \ldots$ have the properties (i)–(iv) (with V_0 replaced by V_1). We also have that the set of functions g_j with $NF_1(g_j^q) = 0$ forms an $\mathbb{F}_q$-basis for DV_1. Observe that Theorem 8 is still true with NF_1 replacing NF_0 and V_2 replacing V_1.

Continuing in this fashion we obtain $\mathbb{F}_q$-bases $g_1^{(i)}, g_2^{(i)}, \ldots \in DV_i$ for DV_i for $i = 1, 2, \ldots$. As above, we let $\mathcal{P}_i$ the set of all pole numbers at Q_∞ of the functions $D^{q-1} g_1^{(i)}, D^{q-1} g_2^{(i)}, \ldots$. Then there exist unique $a_i \in \mathbb{F}_q$ and $r \in \mathbb{F}_q[x, y]$ such that

$$g = \sum_j a_j D^{q-1} g_j^{(i)} + r$$

where the pole order every term of $r \in \mathbb{F}_q[x, y]$ does not belong to $\mathcal{P}_i$. This follows since the functions $D^{q-1} g_j^{(i)}$ $(j \geqslant 1)$ have distinct pole orders. We denote the function r by $NF_i(g)$ and we refer to r as the normal form of g *with respect to* $g_1^{(i)}, g_2^{(i)}, \ldots$. Note that NF_i is an $\mathbb{F}_q$-linear map. In Corollary 5 it is shown that the sequence $V_1, V_2, \ldots$ must stabilize at the desired integral closure. Note that $\mathcal{P}_0 \supseteq \mathcal{P}_1 \supseteq \mathcal{P}_2 \supseteq \cdots$ and, from the Weierstrass Gap Theorem [11], every integer $\geqslant (q-1)db + 2g - 2$ belongs to $\mathcal{P}_0$.

In order to find a basis of the finite dimensional space $\mathcal{L}(mQ_\infty)$, it is not necessary to use all of the functions $g_i^{(0)}$ $(i \geqslant 1)$. In Proposition 13, we show that round 1 should begin with $K(q^t m + bd)$ functions (see (8) for a formula).

(a) **Input: A prescribed integer m.**

(b) **Initialization: Arrange the monomials $x^i y^j$ $(0 \leqslant j \leqslant b-1$ and $i = 0, 1, 2, \ldots)$ in increasing pole order. Denote these as $g_1^{(0)}, g_2^{(0)}, \ldots$. Put $t := \lceil \log_q(b\ell) \rceil + 1$ and $S_i = \emptyset$ for $i = 1, 2, 3, \ldots, t$.**

(c) **Procedure Adjust(S_ι)**

For i from 1 to $\#S_\iota$ do

$r_i := NF_\iota(g_i^{q^t})$ **(computed with respect to the functions in $S_{\iota-1}$ if $\iota > 1$ otherwise with respect to the functions $g_1^{(0)}, g_2^{(0)}, \ldots$)**

Let $\mathrm{LT}(r_i) = a_i x^{b_i} y^{c_i}$ **and so** $\mathrm{LM}(r_i) = x^{b_i} y^{c_i}$

Repeat the following operation for as long as possible:

For $1 \leqslant j < i$**, if** $\mathrm{LM}(r_i) = \mathrm{LM}(r_j)$ **then replace** r_i **by** $r_i - \frac{a_i}{a_j} r_i$ **and replace** g_i **by** $g_i - \frac{a_i}{a_j} g_j$**.**
If $r_i = 0$ **then append** g_i **to** $S_{\iota+1}$ **and exit procedure.**
end i **loop.**

(d) **Put** $g_i := g_i^{(0)}$ **for all** i**,** $S_1 := (g_1, g_2, \ldots, g_K)$ **and** $\iota := 1$

(1) Repeat the following for as long as possible:
For i **from** 1 **to** t **do Adjust**(S_i)**.**

(e) **If** $S_t = \{g_1, g_2, \ldots, g_\ell\}$ **(the** g_i**'s written in increasing pole order) then** S **is a basis for** $\mathcal{L}(m' Q_\infty)$ **where** $m' \geqslant m$ **is the pole order of** g_ℓ**.**

In step (e) above, the algorithm can be halted as soon as a g_i in S_t has pole order $m + bd$ (or $m + 1 + bd$ if m happens to be a gap number). Below, in Section 4, we illustrate this algorithm with an example.

Using a standard procedure, the above algorithm can be used to find an $\mathbb{F}_q$-basis for $\mathrm{ic}(R)$ in a finite number of steps. Suppose we have a basis $f_1, f_2, \ldots, f_t$ for $L((2g-1)Q_\infty)$. By Riemann–Roch Theorem, $t = 2g - 1 - g + 1 = g$. By the Weierstrass Gap Theorem [11], the numbers $2g, 2g+1, \ldots$ are pole numbers. Choose functions f_{g+i} with pole number $2g + i - 1$ for $i = 1, 2, \ldots, b$. So $f_{g+i} \in \mathcal{L}((2g + i - 1)Q_\infty)$ for $i = 1, 2, \ldots, b$.

Theorem 9. *The functions* $f_1, f_2, \ldots, f_g, f_{g+1}, f_{g+2}, \ldots, f_{g+b}, xf_{g+1}, xf_{g+2}, \ldots, xf_{g+b}, x^2 f_{g+1}, x^2 f_{g+2}, \ldots, x^2 f_{g+b}, \ldots$ *form an* $\mathbb{F}_q$*-basis for* $\mathrm{ic}(R)$.

Proof. This result is standard, see [10] for example. □

Remark 10. The algorithm does apply to a broader class of functions fields. Define the function field $F := \mathbb{F}_q(x, y)$ by

$$f(y) = y^b + a_1 y^{b-1} + a_2 y^{b-2} + \cdots + a_b \tag{7}$$

where $a_i \in \mathbb{F}_q[x]$ and $f(T) = T^b + a_1 T^{b-1} + a_2 T^{b-2} + \cdots + a_b$ is irreducible and separable. We assume that the pole P_∞ of x in $\mathbb{F}_q(x)$ is totally ramified in F. Let Q_∞ denote the unique place of F which lies above P_∞. Since y is integral over $\mathbb{F}_q[x]$, it follows that Q_∞ is a pole of y. The algorithm still applies to F if we assume that the pole order of y at Q_∞ is relatively prime to b.

4. Example

In this section we work a simple example in order to illustrate the algorithm. Consider the function field $F := \mathbb{F}_2(x, y)$ where $f(y) = y^2 + (x+1)y + x^9$ so $a = 9$ and $b = 2$. Here $f'(y) = x + 1$ so $D = D(x) = x + 1$. It follows that $d = \ell = 1$. The smallest integer t such that $2^t > b\ell = 4$ is 2. Thus, from Corollary 5, $V_2 = \mathrm{ic}(\mathbb{F}_2[x])$ and so the algorithm has 2 rounds. In this example, we illustrate the algorithm by computing a basis for $\mathcal{L}(7Q_\infty)$. From (8) one checks that we should begin with $K(2^t m + bd) = K(30) = 30$ functions $g_i^{(0)}$. The functions $g_1^{(0)}, g_2^{(0)}, \ldots, g_{13}^{(0)}$ are listed below.

Remark 11. In practice, we do not work with all of the $K(30) = 30$ functions because the table in round 2 is updated as the functions become available and the necessary computations in round 2

are done with these functions as far as possible, thus minimizing redundant computation. In this example, the desired basis is achieved by the time we use the 13th function in round 1. In general, to streamline computation, tables in subsequent rounds can be updated as soon as possible and the relevant computations for those rounds can be executed as far as possible.

i	1	2	3	4	5	6	7	8	9	10	11	12	13
$g_i^{(0)}$	1	x	x^2	x^3	x^4	y	x^5	xy	x^6	x^2y	x^7	x^3y	x^8
Pole order	0	2	4	6	8	9	10	11	12	13	14	15	16

Thus the pole orders of the functions $D^{q-1}g_i^{(0)}$ are $2, 4, 6, 8$ and $10 \leqslant i \leqslant 18$: these are the initial numbers in the set $\mathcal{P}_0$. We use the following notation to save space: we write $g \sim h$ if $NF_i(g) = NF_i(h)$.

Round 1. Below we compute the normal forms r_i of $(g_i^{(0)})^2$ and then apply Gaussian reduction. Put $g_i := g_i^{(0)}$ for all i.

- $g_1^2 = 1$, so $r_1 = 1$.
- $g_2^2 = x^2 \sim x^2 - (x+1)g_2^{(0)} = x \sim x - (x+1)g_1^{(0)} = 1$. So $r_2 = 1$. Applying Gaussian reduction, we get $r_2 := r_2 - r_1 = 0$ and $g_2 := g_2 - g_1 = x + 1$. *The function* $x + 1$ *now qualifies for round two.*
- $g_3^2 = x^4 \sim x^4 - (x+1)g_4^{(0)} = x^4 - (x+1)x^3 = x^3$. Continuing in this way we see that $x^3 \sim 1$ so $r_3 = 1$. Applying Gaussian reduction, we get that $r_3 := r_3 - r_1 = 0$ and $g_3 := g_3 - g_1 = x^2 + 1$. *The function* $x^2 + 1$ *qualifies for round two.* In general for any integer $n > 1$ we have $x^n \sim x^n - (x+1)x^{n-1} = x^{n-1}$ whence $x^n \sim 1$ so the functions $x^n + 1$ qualify for round 2.
- $g_6^2 = y^2 = (x+1)y + x^9 \sim x^9 \sim 1$. Applying Gaussian reduction we get $r_6 := r_6 - r_1 = 0$ and $g_6 := g_6 - g_1 = y + 1$. *The function* $y + 1$ *qualifies for round two.*
- $g_8^2 = x^2y^2 = x^2((x+1)y + x^9) = (x+1)x^2y + x^{11} \sim x^{11} \sim 1$. Applying Gaussian reduction we get $r_8 := r_8 - r_1 = 0$ and $g_8 := g_8 - g_1 = xy + 1$. *The function* $xy + 1$ *qualifies for round two.* In general, $(x^n y)^2 = x^{2n}y^2 = x^{2n}((x+1)y + x^9) = (x+1)x^{2n}y + x^{2n+9} \sim 1$. *The functions* $x^n y + 1$ *qualify for round two.*

We have accounted for all of the entries in the table below.

Round 2. The functions $g_i^{(1)}$ together with their pole orders are listed below:

i	1	2	3	4	5	6	7	8	9	10	11	12
$g_i^{(1)}$	$x+1$	x^2+1	x^3+1	x^4+1	$y+1$	x^5+1	$xy+1$	x^6+1	x^2y+1	x^7+1	x^3y+1	x^8+1
Pole order	2	4	6	8	9	10	11	12	13	14	15	16

The pole orders of the functions $D^{q-1}g_i^{(1)}$ are $4, 6, 8, 10, 11, 12, \ldots$. These are the initial numbers in $\mathcal{P}_1$. Below we compute the normal forms of $(g_i^{(1)})^2$ with respect to the $g_i^{(1)}$ and apply Gaussian reduction. Put $g_i := g_i^{(1)}$ for $1 \leqslant i \leqslant 12$.

- $g_1^2 = (x+1)^2 \sim (x+1)^2 - (x+1)g_1^{(1)} = 0$. Thus $r_1 = 0$. *The function* $(x+1)/D = 1$ *is the first basis element of* $\mathcal{L}(7Q_\infty)$.
- $g_2^2 = x^4 + 1 = x^4 + 1 - (x+1)g_4^{(1)} \sim x^4 + 1 - (x+1)(x^3+1) = x + x^3 x + x^3 - (x+1)(x^2+1) = 1 + x^2 \sim 1 + x^2 - (x+1)\cdot(x+1) = 0$. Thus $r_2 = 0$. *The function* $(x^2+1)/D = x+1$ *is the next basis element of* $\mathcal{L}(7Q_\infty)$. Note that in general for $n \geqslant 1$, $x^n = x^n + 1 - (x+1)(x^{n-1}+1) = x^n + 1 + x^n + x + x^{n-1} + 1 = x^{n-1} + x + 1 \sim x^{n-1} + x + 1 - (x+1)(x^{n-2}+1) = x^{n-1} + x + 1 + x^{n-1} + x + x^{n-2} + 1 = x^{n-2}$, so $x^n \sim 1$ if n even and $x^n \sim x$ if n is odd.
- $g_3^2 = x^6 + 1 \sim 1 + 1 = 0$. So $r_3 = 0$. *The function* $(x^3+1)/D = x^2 + x + 1$ *is the next basis element of* $\mathcal{L}(7Q_\infty)$.
- $g_4^2 = x^8 + 1 \sim 1 + 1 = 0$. So $r_4 = 0$. *The function* $(x^4+1)/D = (x+1)^3$ *is the next basis element of* $\mathcal{L}(7Q_\infty)$.
- $g_5^2 = y^2 + 1 = (x+1)y + x^9 + 1 = (x+1)(y+1) + x + 1 + x^9 + 1 \sim x + x^9 \sim x + x = 0$, so $r_5 = 0$. *The function* $(y+1)/D = (y+1)/(x+1)$ *is the next basis element of* $\mathcal{L}(7Q_\infty)$.

The pole order of $(y+1)/(x+1)$ is 7 so we can now stop. The computed basis for $\mathcal{L}(7Q_\infty)$ is $1, x+1, x^2+1, x^2+x+1, (x+1)^3, (y+1)/(x+1)$.

5. Complexity

In this section we determine the complexity of the algorithm described in Section 3. Throughout, whenever we speak of the pole order of a function in F, it is understood to be the pole order with respect to the place Q_∞.

Lemma 12.

(1) *Let* $M \geqslant 1$. *Then the number of functions* $x^i y^j$ $(i \geqslant 0,\ 0 \leqslant j \leqslant b-1)$ *with pole order* $bi + aj \leqslant M$ *is*

$$K(M) := \sum_j d + 1 + \left\lfloor \frac{M - aj}{b} \right\rfloor \leqslant M + 1 \tag{8}$$

where the sum is over all j *such that* $0 \leqslant j \leqslant \min(\lfloor \frac{bd+M}{a} \rfloor, b-1)$.

(2) *The number of functions* $g_i^{(j)}$ $(i, j \geqslant 0)$ *with pole order* $\leqslant M$ *is at most* $K(M)$.

Proof. Since $ib + ja \leqslant M$, it follows that $i \leqslant d + \lfloor \frac{M-aj}{b} \rfloor$. If $0 \leqslant j \leqslant b-1$ is fixed and $d + (M - aj)/b \geqslant 0$, that is, $j \leqslant \lfloor \frac{bd+M}{a} \rfloor$ then the number of possible values for i is at most $d + 1 + \frac{M-aj}{b}$. Thus total number of i, j such that $ib + ja \leqslant M$ is

$$K(M) := \sum_j d + 1 + \left\lfloor \frac{M - aj}{b} \right\rfloor \tag{9}$$

where the sum is over all j such that $0 \leqslant j \leqslant \min(\lfloor \frac{bd+m_0}{a} \rfloor, b-1)$. Since $K(M)$ counts the number of nonnegative integers $\leqslant$ representable in the form $bi + aj$ and some of these numbers may not have such a representation, the inequality in (8) follows.

The second result (2) now follows since the pole order of any $g_i^{(k)}$ is of the form $bi + aj$ for some $i \geqslant 0$ and $0 \leqslant j \leqslant b-1$. □

Proposition 13. *Put $m_j := q^{t-j}m + bd$ for $0 \leqslant j \leqslant t$. In order to obtain a basis for $\mathcal{L}(mQ_\infty)$, the algorithm should begin with at most $K(m_0)$ of the functions $g_i^{(0)}$. These functions have pole order at most m_0. We compute the normal forms of the qth powers of $K(m_1)$ of those functions among the $g_i^{(0)}$ with pole orders at most m_1.*

For $1 \leqslant j \leqslant t$, the functions $g_i^{(j-1)}$ have pole orders $\leqslant m_{j-1}$ and there are at most $K(m_{j-1})$ such functions. For $j < t$, we compute the normal forms of the qth powers of those functions among the $g_i^{(j-1)}$ with pole orders at most m_j and there are at most $K(m_j)$ such functions.

Proof. Note that in the jth round of the algorithm, the functions $g_i^{(j-1)}$ are used to compute the functions $g_i^{(j)}$. In the final round (that is round t) the functions $g_i^{(t)}$ are output with pole orders at most $m + bd$ (since the function $g_i^{(t)}/D$ must have pole order $\leqslant m$). In order to obtain these functions, we had to compute the normal forms of the qth powers of only those $g_i^{(t-1)}$ (with respect to $g_1^{(t-1)}, g_2^{(t-1)}, \ldots$) with pole orders $\leqslant m + bd$. In order to have computed these normal forms, we require functions $D^{q-1}g_i^{(t-1)}$ with pole orders $\leqslant q(m + bd)$, that is, the functions $g_i^{(t-1)}$ must have pole order $\leqslant q(m + bd) - (q-1)bd = qm + bd$. Thus, in round t, we need functions $g_i^{(t-1)}$ with pole order $\leqslant qm + bd = m_{t-1}$ and we compute the normal forms of the qth powers of only those with pole order $\leqslant m + bd = m_t$.

Likewise, since the functions $g_i^{(t-1)}$ must have pole orders $\leqslant qm + bd$, in round $t-1$, we need functions $g_i^{(t-2)}$ with pole orders $\leqslant q^2m + bd = m_{t-2}$. Also we compute the normal forms of qth powers of the functions $g_i^{(t-2)}$ with pole order at most $qm + bd = m_{t-1}$.

In general, the functions $g_i^{(j)}$ must have pole orders $\leqslant q^{t-j}m + bd$ and so in the jth round, we require that the functions $g_i^{(j-1)}$ have pole orders at most $q^{t-j+1}m + bd = m_{j-1}$. We compute the normal forms of the qth powers of the functions $g_i^{(j)}$ with pole orders at most $q^{t-j}m + bd = m_j$.

Thus, in round 1, we require functions $g_i^{(0)}$ with pole orders $\leqslant q^t m + bd = m_0$. We compute the normal forms of the qth powers of the functions $g_i^{(0)}$ with pole orders at most $q^{t-1}m + bd$.

The rest of Proposition 13 follows from Lemma 12. □

We will use several results on complexity from computer algebra. As a general reference we use [1]. For example, the complexity of multiplying two polynomials in $\mathbb{F}_q[x]$ of degree at most n is $O(n^2)$ [1, Chapter 2]. If M is an $r \times n$ matrix with entries from $\mathbb{F}_q$ and $r \gg n$ then the complexity of Gaussian reduction to transform M into an upper triangular matrix is $O(rn^2)$.

From Riemann's Inequality [11], the genus g of F satisfies $g \leqslant (a-1)(b-1) \leqslant ab$ so

$$g \in O(ab). \tag{10}$$

Observe that $K(m_j) \leqslant m_j + 1 \leqslant m_0 + 1$ and $m_0 = q^t m + bd$. Now t is the smallest integer such that $q^t > b\ell$ so $q^{t-1} \leqslant b\ell \leqslant bd$ and $m_0 \leqslant qb\ell m + bd$. Thus $m_0 \in O(qbdm)$ and we also have $K(m_j) \in O(qbdm)$.

Theorem 14. *The complexity of the qth power algorithm to compute a basis of* $\mathcal{L}(mQ_\infty)$ *is*

$$\mathrm{O}\big(q^3a^3b^9m^3\log_q\big(ab^3\big)\big). \tag{11}$$

Consequently, the complexity to compute an $\mathbb{F}_q$ *basis for the integral closure is*

$$\mathrm{O}\big(q^3a^6b^{12}\log_q\big(qb^3\big)\big).$$

Proof. We compute the complexity of the different parts of the algorithm separately:

1) *The cost of computing the polynomial* $D(x)$ *is* $\mathrm{O}(a^2b^5)$ *and*

$$d = \deg D(x) = \mathrm{O}\big(ab^2\big). \tag{12}$$

From (6) we have that $y^b = -x^a - g(x,y) = c_0 + c_1y + \cdots + c_{b-1}y^{b-1}$ where $c_0, c_1, \ldots, c_{b-1} \in \mathbb{F}_q[x]$ have maximum degree a. By using this relation, for $j > b$ the function y^j can be computed as a linear combination of $1, y, \ldots, y^{b-1}$ with coefficients in $\mathbb{F}_q[x]$. To find $D(x)$ one has to solve the following equation:

$$1 = f'(y)\big(x_0 + x_1y + \cdots + x_{b-1}y^{b-1}\big) \tag{13}$$

for $x_0, x_1, \ldots, x_{b-1}$. Eq. (13) can be written in the matrix form $AX = C$ where A is a $b \times b$ matrix with coefficients in $\mathbb{F}_q[x]$ and C is the transpose of $[1, 0, \ldots, 0]$. The polynomial D divides the determinant of A. The cost of computing the entries of the matrix A is at most $b(a^2 + (2a)^2 + \cdots + ((b-1)a)^2)\mathrm{O}(1) \in \mathrm{O}(a^2b^4)$. Note the maximum degree of the polynomials in the ith row of A is at most ia. So the cost [9] to compute the determinant of A is $\mathrm{O}(b^3(ba)^2) = \mathrm{O}(a^2b^5)$. Thus the overall complexity to compute $D(x)$ is $\mathrm{O}(a^2b^5)$ and $d = \deg D(x) \leqslant ab(b-1)$ whence $d \in \mathrm{O}(ab^2)$.

Since $t \in \mathrm{O}(\log_q(bd))$ we see that

$$t \in \mathrm{O}\big(\log_q\big(ab^3\big)\big). \tag{14}$$

Thus

$$m_0 \in \mathrm{O}\big(qab^3m\big) \tag{15}$$

and we also have

$$K(m_j) \in \mathrm{O}\big(qab^3m\big). \tag{16}$$

2) *The total cost of Gaussian elimination on the* r_i *and the* $(g_i^{(j)})^q$ *is* $\mathrm{O}(q^3a^3b^9m^3\log_q(ab^3))$. From the Weierstrass Gap Theorem, the pole orders of the r_i is at most $(q-1)d + 2g - 1$. In the jth round the total cost of Gaussian elimination is: $\mathrm{O}(K(m_j)[(q-1)d + 2g - 1]^2) = \mathrm{O}(m_0[(q-1)d + 2g - 1]^2)$ (since $K(m_j) \leqslant m_j + 1 \leqslant m_0 + 1$). From (16), (12), (10) and the fact that there are t rounds the total cost of Gaussian elimination on the r_i for the entire algorithm is $\mathrm{O}(t(qab^3m)\cdot(qab^2)^2) = \mathrm{O}((qab^3m)\cdot(qab^2)^2\log_q(ab^3))$, that is,

$$\mathrm{O}\big(q^3a^2b^5m\log_q\big(ab^3\big)\big). \tag{17}$$

Next we determine the cost of computing the g_i^q in any round. In the first round one computes $(x^iy^j)^q$. The quantities $g_i^{(\iota)}$ can be computed in subsequent rounds by exploiting the linearity of the qth power, that is, by modifying Gaussian reduction to:

Gaussian reduction. *If the pole order of r_i equals that of r_j for any $1 \leqslant j < i$, then replace r_i by $r_i - \frac{a_i}{a_j}r_j$ and replace g_i by $g_i - \frac{a_i}{a_j}g_j$ and replace g_i^q by $g_i^q - \frac{a_i}{a_j}g_j^q$.*

First we determine the cost of computing the $(x^iy^j)^q$ for $1 \leqslant j \leqslant b-1$ and $i \geqslant 0$. Observe that $(x^iy^j)^q = x^{iq}y^{jq}$. It suffices to determine the cost of computing y^{jq} $(1 \leqslant j \leqslant b-1)$. Now $y^b = c_0 + c_1y + \cdots + c_{b-1}y^{b-1}$ so $y^{b+1} = c_0y + c_1y^2 + \cdots + c_{b-2}y^{b-1} + c_{b-1}(c_0 + c_1y + \cdots + c_{b-1}y^{b-1})$. To determine the coefficients of the y^i $(1 \leqslant i \leqslant b-1)$, we need to multiply two polynomials of degree at most a. Thus the complexity to compute y^{b+1} is $b\mathrm{O}(a^2)$. Thus the total cost to compute y^j $(b+1 \leqslant j \leqslant (b-1)q)$ by computing successive powers of y^j is at most $b\mathrm{O}([(b-1)qa - ba]^3) = b^4a^3q^3\mathrm{O}(1)$.

The complexity of computing the $g_i^{(j)}$ from the qth powers of the $g_i^{(j-1)}$ in round j is $\mathrm{O}(m_j[qm_{j-1}]^2) = \mathrm{O}(q^2m_0^3) = \mathrm{O}(q(ab^3m)^3) = \mathrm{O}(qa^3b^9m^3)$ and the complexity over t rounds is $\mathrm{O}(tqa^3b^9m^3)$, that is,

$$\mathrm{O}\big(q^3a^3b^9m^3\log_q\big(ab^3\big)\big) \tag{18}$$

which dominates $b^4a^3q^3\mathrm{O}(1)$.

3) *The cost of computing $D^{q-1}(x^iy^j)$ is* $\mathrm{O}(qa^2b^4)$. This amounts to the cost of computing D^{q-1}. One can achieve this by first computing D^q and then dividing by D. Thus the cost [1] is $\mathrm{O}(d^2q) = \mathrm{O}(qa^2b^4)$.

4) *The cost of computing the $NF_\iota((g_i^{(\iota)})^q)$ is* $\mathrm{O}(q^3a^3b^9m^3\log_q(ab^3))$. First observe that if $\alpha_i \in \mathbb{F}_q[x]$ and $g = \alpha_0 + \alpha_1y + \cdots + \alpha_{b-1}y^{b-1}$ has pole order $\leqslant K$ then $b\deg\alpha_i + ai \leqslant K$ and $\deg c_i \leqslant K/b$. Thus the total 'length' of g is $\sum_{i=0}^{b-1}\alpha_i \leqslant K$. From Proposition 13, in round j, we compute the normal form of the qth powers of $K(m_{j-1}) \leqslant m_{j-1}+1$ functions, each with pole order at most $(q-1)bd + m_{j-1}$. Thus the cost of Gaussian reduction in round j is

$$\begin{aligned}
\big((q-1)bd + m_{j-1}\big)^2m_{j-1}\mathrm{O}(1) &= \big(qbab^2 + m_0\big)^2m_0\mathrm{O}(1)\\
&= \big(qab^3 + qab^3m\big)^2qab^3m\mathrm{O}(1)\\
&= \big(qab^3m\big)^2qab^3m\mathrm{O}(1)\\
&= \big(qab^3m\big)^2qab^3m\mathrm{O}(1)\\
&= \mathrm{O}\big(q^3a^3b^9m^3\big).
\end{aligned}$$

Since there are t rounds, the total complexity is $\mathrm{O}(q^3a^3b^9m^3\log_q(ab^3))$.

The complexity of the entire algorithm follows now from comparing the complexities of the different parts above.

Next we determine the complexity of computing the integral closure. In order to compute the integral closure, one needs to compute a basis for $\mathcal{L}((2g-1)Q_\infty)$, that is, we take $m = 2g-1 \in \mathrm{O}(ab)$. The complexity for this is $\mathrm{O}(q^3a^6b^{12}\log_q(qb^3))$. □

Acknowledgment

The authors thank one of the referees whose suggestions lead to substantial improvements in this paper.

References

[1] J. von zur Gathen, J. Gerhard, Modern Computer Algebra, Cambridge Univ. Press, New York, 1999, xiv+753 pp.
[2] A. Garcia, H. Stichtenoth, A tower of Artin–Schreier extensions of function fields attaining the Drinfeld–Vladut bound, Invent. Math. 121 (1995) 211–222.
[3] A. Garcia, H. Stichtenoth, On the asymptotic behavior of some towers of function fields over finite fields, J. Number Theory 61 (1996) 248–273.
[4] T. Hoeholdt, J.H. van Lint, R. Pellikaan, Algebraic geometry codes, in: V.S. Pless, W.C. Huffman, R.A. Brualdi (Eds.), Handbook of Coding Theory, vol. 1, Elsevier, Amsterdam, 1998, pp. 871–961.
[5] S. Lang, Algebraic Number Theory, Springer, New York, 1994.
[6] D. Leonard, Finding the defining functions for one-point algebraic–geometry codes, IEEE Trans. Inform. Theory 47 (6) (2001) 2566–2573.
[7] D. Leonard, R. Pellikaan, Integral closure and weight functions over finite fields, Finite Fields Appl. 9 (2003) 479–504.
[8] E. Mattig, Ein Algorithmmus zur Berechnung von Ganzheitsbasen in algebraischen Funktionenkörpern, PhD thesis, Universität Duisburg-Essen, 2003.
[9] T. Mulders, A. Storjohann, On lattice reduction for polynomial matrices, Technical Report 356, Departement Informatik, ETH Zürich, December 2000.
[10] K.W. Shum, I. Aleshnikov, V.P. Kumar, H. Stichtenoth, V. Deolalikar, A low-complexity algorithm for the construction of algebraic–geometric codes better than the Gilbert–Varshamov bound, IEEE Trans. Inform. Theory 47 (6) (2001) 2225–2241.
[11] H. Stichtenoth, Algebraic Function Fields and Codes, Springer, Berlin, 1993.
[12] J.J. Sylvester, Question 7382, Mathematical Questions from the Educational Times 37 (1884) 26.